UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/682,133	10/09/2003	Richard D. Dettinger	ROC920030237US1	1361

7590   05/22/2006

William J. McGinnis, Jr.
IBM Corporation, Dept. 917
3605 Highway 52 North
Rochester, MN 55901-7829

EXAMINER
EBIRIM, EMEKA

ART UNIT	PAPER NUMBER
2166	

DATE MAILED: 05/22/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

	Application No.	Applicant(s)
Office Action Summary	10/682,133	DETTINGER ET AL.
	Examiner	Art Unit	
	Emeka Ebirim	2166	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on <u>10/09/2003</u>.
2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.
3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) <u>1-54</u> is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.
6)☒ Claim(s) <u>1-54</u> is/are rejected.
7)☐ Claim(s) _____ is/are objected to.
8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.
10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☒ Notice of References Cited (PTO-892)
2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3)☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
    Paper No(s)/Mail Date <u>10/26/04, 10/09/03</u>.

4)☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____.
5)☐ Notice of Informal Patent Application (PTO-152)
6)☐ Other: _____.

DETAILED ACTION

Claim Status

1.   Claims 1 – 54 are pending in this Office action.

The application has been examined. Claims 1-54 are rejected as detailed below and are

pending in this office action.

Claim Rejections - 35 USC § 101

2.   35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of
> matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the
> conditions and requirements of this title.

Claims 26 – 39 are rejected under 35 U.S.C. 101 because the claimed invention

is directed to non-statutory subject matter.

Claims 26 – 39 are not limited to tangible embodiments. In view of Applicants'

disclosure, specification page 12 , paragraph 0036, the medium in not limited to tangible

embodiments, instead being defined as including both tangible embodiments (e.g., [CD-

ROM disks]) and intangible embodiments (e.g., [wireless communications]).

As such, the claim is not limited to statutory subject matter and is therefore non-

statutory.

Claim Rejections - 35 USC § 102

3.   The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that

form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

4.   Claims 1-6, 18-20, 23, 26-30, 42-44, 47, 50, 52 are rejected under 35

U.S.C. 102(b) as being anticipated by Pat No: 6,434,545 to MacLeod et al (hereinafter

MacLeod).

Claim 1.

MacLeod discloses:

A method of managing execution of query operations in a data processing

system, comprising [see Abstract]:

issuing, by a requesting entity, a request to perform a composite query operation

defining an initial query operation and a subsequent query operation to be executed

against a data repository of the data processing system [Col 1 lines 65-67, Col 7 lines

28-35, 43-45, Fig 10];

executing the initial query operation [Col 7 lines 43-45, Fig 10];

determining an operation status of the initial query operation [Col 7 lines 49-51,

Fig 10]; and

managing execution of the subsequent query operation on the basis of the

determined operation status [second query (subsequent query) Col 7 lines 63-65, Col 8

lines 1-5,15-25, Fig 5, Fig 10].

Claim 2.

MacLeod discloses the elements of Claim 1 as above and furthermore it

discloses wherein the determining and managing are performed by a composite query

operations manager [Col 6 lines 62-67, Fig 10].

Claim 3.

MacLeod discloses the elements of Claim 1 as above and furthermore it discloses wherein the requesting entity is an application and wherein the determining and managing are performed by a composite query operations manager [application, Col 7 lines 32-36].

Claim 4.

MacLeod discloses the elements of Claim 1 as above and furthermore it discloses wherein the initial and the subsequent query operation are SQL statements [SQL, Col 6 lines 49-53].

Claim 5.

MacLeod discloses the elements of Claim 1 as above and furthermore it discloses wherein determining an operation status of the initial query operation comprises determining a number of items affected by the initial query operation [Col 9 lines 16-17, Fig 11].

Claim 6.

MacLeod discloses the elements of Claim 1 as above and furthermore it discloses wherein determining an operation status of the initial query operation comprises determining whether the initial query operation completed successfully [resolve, checking, Col 6 lines 49-53, 55-60, Fig 4].

Claim 18.

MacLeod discloses

A method of managing execution of query operations in a data processing

system, comprising [see Abstract]:

providing a composite query operation to be executed against a data repository

of the data processing system [Col 1 lines 65-67, Col 7 lines 28-35, 43-45, Fig 10];

providing at least two implementation schemas for the composite query

operation, each defining a different order of execution of at least two different query

operations required to perform the composite query operation [execution plans

(implementation schema), Col 1 lines 65-67-Col 2 line 1-5, Col 6 lines 10-15, Fig 4];

executing the composite query operation according to a first implementation

schema of the implementation schemas [implements the selected execution plan, Col 7

lines 5-9, Fig 4-6];

determining, upon completion of the execution of the composite query operation,

a completion status of the composite query operation [Col 7 lines 5-9, lines 13-17, Fig 4-

6]; and

managing a subsequent execution of the composite query operation on the basis

of the determined completion status [second query (subsequent query) Col 7 lines 63-

65, Col 8 lines 1-5,15-25, Fig 5, Fig 10].

Claim19.

MacLeod discloses the elements of claim 18 as above and furthermore it

discloses wherein the completion status indicates a number of query operations

executed as part of the composite query operation [Fig 5-6].

Claim 20.

MacLeod discloses the elements of claim 18 as above and furthermore it

discloses wherein the completion status indicates a number of query operations

executed as part of the composite query operation [Fig 5-6]; and wherein the managing

comprises:

executing the composite query operation according to a second implementation

schema of the at least two implementation schemas, if the number of query operations

exceeds a predetermined threshold [execution plan, minimal number, Col 6 lines 17-25].

Claim 23.

MacLeod discloses the elements of claim 18 as above and furthermore it

discloses wherein the first and second query operations are SQL statements [SQL, Col

6 lines 49-53].

Claim 50.

MacLeod discloses:

A data processing system comprising [processing unit, See MacLeod, Col 7 lines

23-25]:

a data repository [data stored, database, See MacLeod, Col 5 lines 64-66]; and

a composite query operations manager residing in memory for managing

execution of query operations in the data processing system, the composite query

operations manager being configured for [management system, execution plan, See

MacLeod, Col 6 lines 10-15]:

receiving a request to perform a composite query operation defining an initial

query operation and a subsequent query operation to be executed against the data

repository [MacLeod, Col 1 lines 65-67, Col 7 lines 28-35, 43-45, Fig 10];

executing the initial query operation [execution plan, See MacLeod, Col 6 lines

10-15];

determining an operation status of the initial query operation [Col 7 lines 49-51,

Fig 10]; and

managing execution of the subsequent query operation on the basis of the

determined operation status [second query (subsequent query) Col 7 lines 63-65, Col 8

lines 1-5,15-25, Fig 5, Fig 10].

Claim 52.

MacLeod discloses:

A data processing system comprising [processing unit, See MacLeod, Col 7 lines

23-25]:

a data repository [data stored, database, See MacLeod, Col 5 lines 64-66]; and

a composite query operations manager residing in memory for managing

execution of query operations in the data processing system, the composite query

operations manager being configured for [management system, execution plan, See

MacLeod, Col 6 lines 10-15]:

receiving a request to perform a composite query operation against a data

repository of the data processing system [MacLeod, Col 1 lines 65-67, Col 7 lines 28-

35, 43-45, Fig 10];

providing at least two implementation schemas for the composite query

operation, each defining a different order of execution of at least two different query

operations required to perform the composite query operation [execution plans

(implementation schema), Col 1 lines 65-67-Col 2 line 1-5, Col 6 lines 10-15, Fig 4];

executing the composite query operation according to a first implementation

schema of the implementation schemas [implements the selected execution plan, Col 7

lines 5-9, Fig 4-6];

determining, upon completion of the execution of the composite query operation,

a completion status of the composite query operation [Col 7 lines 5-9, lines 13-17, Fig 4-

6]; and

managing a subsequent execution of the composite query operation on the basis

of the determined completion status [second query (subsequent query) Col 7 lines 63-

65, Col 8 lines 1-5,15-25, Fig 5, Fig 10].

Subject matters of claims 26-30, 42-44, 47, are rejected in the analysis above in

claims 1-6, 18-20, 23 and these claims are rejected on that basis.

Claim Rejections - 35 USC § 103

5.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set
> forth in section 102 of this title, if the differences between the subject matter sought to be patented and
> the prior art are such that the subject matter as a whole would have been obvious at the time the
> invention was made to a person having ordinary skill in the art to which said subject matter pertains.
> Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

6.   Claims 7-8, 21-22, 24-25, 31-32, 45-46, 48-49, 53 are rejected under 35 U.S.C. 103(a) as being unpatentable over MacLeod in view of Pat No: 5,412,804 to Krishna (hereinafter Krishna).

Claim 7.

MacLeod discloses the elements of Claim 1 as above and furthermore it discloses wherein determining an operation status of the initial query operation comprises determining, on the basis of a return code received upon completion of the initial query operation, whether the initial query operation completed successfully [See MacLeod Col 6 lines 50-57].

MacLeod discloses the elements of claim 7 as above but it does not explicitly indicate "return code". Krishna discloses the claimed "return code" (both successful and unsuccessful execution of a query ends with a return) [see Krishna Fig 3, Col 6 lines 40-45].

It would have been obvious to one of ordinary skill in the art of data processing to have combined the cited references because "return code" as disclosed by Krishna would have enabled MacLeod to provide an alternative for un-nesting queries with equi-join predicate and a "COUNT" aggregate between the nested blocks [Krishna Col 3 lines 19-22].

Furthermore it would enable integrated procedure for evaluating joins and outer joins in a top-down order [Krishna Col 3 lines 37-39]

Claim 8.

MacLeod discloses the elements of Claim 1 as above and furthermore it discloses wherein managing execution of the subsequent query operation comprises:

executing the subsequent query operation only if the initial query operation did not complete successfully [See MacLeod Col 6 lines 50-57].

MacLeod discloses the elements of claim 8 but did not explicitly indicate "did not complete successfully" Krishna discloses the claimed invention (In the case of unsuccessful execution, another submission is made) [return with error (did not complete successfully), see Krishna Fig 3, Col 6 lines 40-45].

It would have been obvious to one of ordinary skill in the art of data processing to have combined the cited references because "did not complete successfully" as disclosed by Krishna would have enabled MacLeod to provide an alternative for un-nesting queries with equi-join predicate and a "COUNT" aggregate between the nested blocks [Krishna Col 3 lines 19-22].

Furthermore it would enable integrated procedure for evaluating joins and outer

joins in a top-down order [Krishna Col 3 lines 37-39]

Claim 21

MacLeod discloses the elements of claim 18 as above and furthermore it

discloses wherein the completion status indicates a number of query operations

executed as part of the composite query operation [See MacLeod, Fig 5-6]; and wherein

the managing comprises:

repeatedly executing the composite query operation in response to subsequent

requests for execution of the composite query operation according to the first

implementation schema [repeat, execution plan, See MacLeod , Col 7 lines 62-65, Fig

10]; and

upon each execution of the composite query operation according to the first

implementation schema [See MacLeod Col 6 lines 10-15]:

executing the composite query operation according to a second implementation

schema of the implementation schemas [repeat, new execution plan, See MacLeod Col

8 lines 1-6, Fig 10].

MacLeod discloses the elements of claim 21 as above but it does not explicitly

indicate "count". Krishna discloses the claimed "count" [see Krishna Fig 8-15, Col 10

lines 24-25, 37-39, Col 16 lines 5-10, Col].

It would have been obvious to one of ordinary skill in the art of data processing to

have combined the cited references because "count" as disclosed by Krishna would

have enabled MacLeod to provide an alternative for un-nesting queries with equi-join

predicate and a "count" aggregate between the nested blocks [Krishna Col 3 lines 19-22].

Furthermore it would enable integrated procedure for evaluating joins and outer joins in a top-down order [Krishna Col 3 lines 37-39].

Claim 22.

MacLeod discloses the elements of claim 18 as above and furthermore it discloses wherein the completion status indicates a number of query operations executed as part of the composite query operation [See MacLeod Fig 5-6]; and wherein the managing comprises:

repeatedly executing the composite query operation in response to subsequent requests for execution of the composite query operation according to the first implementation schema [repeat, execution plan, See MacLeod Col 7 lines 62-65, Fig 10]; and

upon each execution of the composite query operation according to the first implementation schema [See MacLeod Col 6 lines 10-15]:

summing the number of query operations indicated by the completion status to the accumulated number of query operations [See MacLeod Col 9 lines 23-28]; and

if the accumulated number of query operations exceeds a predetermined threshold [execution plan, minimal number, See MacLeod Col 6 lines 17-25]:

executing the composite query operation according to a second implementation

schema of the implementation schemas [repeat, new execution plan, See MacLeod Col

8 lines 1-6, Fig 10].

MacLeod discloses the elements of claim 22 as above but it does not explicitly

indicate "accumulated number". Krishna discloses the claimed "accumulated number"

(count) [see Krishna Fig 8-15, Col 10 lines 24-25, 37-39, Col 16 lines 5-10, Col].

It would have been obvious to one of ordinary skill in the art of data processing to

have combined the cited references because "count" as disclosed by Krishna would

have enabled MacLeod to provide an alternative for un-nesting queries with equi-join

predicate and a "count" aggregate between the nested blocks [Krishna Col 3 lines 19-

22].

Furthermore it would enable integrated procedure for evaluating joins and outer

joins in a top-down order [Krishna Col 3 lines 37-39].

Claim 24.

A method of managing execution of query operations in a data processing

system, comprising [see Abstract]:

a) providing a composite query operation defining a first and a second query

operation to be executed against a data repository of the data processing system [Col 1

lines 65-67, Col 7 lines 28-35, 43-45, Fig 10];

b) providing a first and a second implementation schema for the composite query

operation, each defining a different order of execution of the first and the second query

operation [execution plans (implementation schema), Col 1 lines 65-67-Col 2 line 1-5,

Col 6 lines 10-15, Fig 4];

c) selecting an implementation schema of the first and second implementations

schemas [select execution plans (implementation schema), Col 1 lines 65-67-Col 2 line

1-5, Col 6 lines 10-15, Fig 4];

e) repeatedly executing the composite query operation according to the selected

implementation schema in response to subsequent requests for execution of the

composite query operation according to the selected implementation schema [repeat,

execution plan, See MacLeod , Col 7 lines 62-65, Fig 10]; and

f) upon each execution of the composite query operation according to the

selected implementation schema [See MacLeod Col 6 lines 10-15]:

f1) determining a completion status of the composite query operation indicating a

number of query operations executed as part of the composite query operation [Col 7

lines 5-9, lines 13-17, Fig 4-6];

f2) incrementing the failure count if the number of query operations indicated by

the completion status exceeds a first predetermined threshold [execution plan, minimal

number (threshold), See MacLeod Col 6 lines 17-25];

f3) determining whether the failure count exceeds a second predetermined

threshold [execution plan, minimal number (threshold), See MacLeod Col 6 lines 17-25];

and

f4) if the failure count exceeds the second predetermined threshold [execution

plan, minimal number (threshold), See MacLeod Col 6 lines 17-25]:

MacLeod discloses the elements of claim 21 as above but it does not explicitly

indicate "count". Krishna discloses the claimed "count" [see Krishna Fig 8-15, Col 10

lines 24-25, 37-39, Col 16 lines 5-10, Col].

It would have been obvious to one of ordinary skill in the art of data processing to

have combined the cited references because "count" as disclosed by Krishna would

have enabled MacLeod to provide an alternative for un-nesting queries with equi-join

predicate and a "count" aggregate between the nested blocks [Krishna Col 3 lines 19-

22].

Furthermore it would enable integrated procedure for evaluating joins and outer

joins in a top-down order [Krishna Col 3 lines 37-39].

Claim 25.

The combination of MacLeod and Krishna discloses the elements of claim 24 as

above and furthermore it discloses, wherein the first and second query operations are

SQL statements [See MacLeod, SQL, Col 6 lines 49-53].

Claim 53.

MacLeod discloses:

A data processing system comprising [processing unit, See MacLeod, Col 7 lines

23-25]:

a data repository [data stored, database, See MacLeod, Col 5 lines 64-66]; and

a composite query operations manager residing in memory for managing

execution of query operations in the data processing system, the composite query

operations manager being configured for [management system, execution plan, See

MacLeod, Col 6 lines 10-15]:

a) receiving a request to perform a composite query operation defining a first and

a second query operation to be executed against a data repository of the data

processing system [Col 1 lines 65-67, Col 7 lines 28-35, 43-45, Fig 10];

b) retrieving a first and a second implementation schema for the composite query

operation, each defining a different order of execution of the first and the second query

operation [execution plans (implementation schema), Col 1 lines 65-67-Col 2 line 1-5,

Col 6 lines 10-15, Fig 4];

c) selecting an implementation schema of the first and second implementations

schemas [select execution plans (implementation schema), Col 1 lines 65-67-Col 2 line

1-5, Col 6 lines 10-15, Fig 4];

e) repeatedly executing the composite query operation according to the selected

implementation schema in response to subsequent requests for execution of the

composite query operation according to the selected implementation schema [repeat,

execution plan, See MacLeod , Col 7 lines 62-65, Fig 10]; and

f) upon each execution of the composite query operation according to the

selected implementation schema [See MacLeod Col 6 lines 10-15]:

f1) determining a completion status of the composite query operation indicating a

number of query operations executed as part of the composite query operation [Col 7

lines 5-9, lines 13-17, Fig 4-6];

f2) incrementing the failure count if the number of query operations indicated by the completion status exceeds a first predetermined threshold [execution plan, minimal number (threshold), See MacLeod Col 6 lines 17-25];

f3) determining whether the failure count exceeds a second predetermined threshold [execution plan, minimal number (threshold), See MacLeod Col 6 lines 17-25]; and

f4) if the failure count exceeds the second predetermined threshold [execution plan, minimal number (threshold), See MacLeod Col 6 lines 17-25]:

MacLeod discloses the elements of claim 21 as above but it does not explicitly indicate "count". Krishna discloses the claimed "count" [see Krishna Fig 8-15, Col 10 lines 24-25, 37-39, Col 16 lines 5-10, Col].

It would have been obvious to one of ordinary skill in the art of data processing to have combined the cited references because "count" as disclosed by Krishna would have enabled MacLeod to provide an alternative for un-nesting queries with equi-join predicate and a "count" aggregate between the nested blocks [Krishna Col 3 lines 19-22].

Furthermore it would enable integrated procedure for evaluating joins and outer joins in a top-down order [Krishna Col 3 lines 37-39].

Subject matters of claims 31-32, 45-46, 48-49, are rejected in the analysis above in claims 7-8, 21-22, 24-25 and these claims are rejected on that basis.

7.   Claims 9-12, 14, 16-17, 33-36, 38, 40-41, 51, 54 are rejected under 35 U.S.C.

103(a) as being unpatentable over MacLeod in view of Pub No: 2002/0078015 to

Ponnekanti (hereinafter Ponnekanti).

Claim 9.

MacLeod discloses:

A method of managing execution of query operations in a data processing

system, comprising [see Abstract]:

issuing, by a requesting entity, a request to perform a composite query operation

defining an initial query operation and a plurality of subsequent query operations to be

executed against a data repository of the data processing system [MacLeod, Col 1

lines 65-67, Col 7 lines 28-35, 43-45, Fig 10];

providing selection logic defining a next query operation of the composite query

operation to be executed [next query operation, MacLeod, Fig 10];

managing, using a composite query operations manager, execution of the initial

query operation and the plurality of subsequent query operations on the basis of the

selection logic and the plurality of failure conditions [query with multiple operations,

select, MacLeod, Fig 5-6, Fig 10, Col 8 lines 31-35].

MacLeod discloses the elements of claim 9 as above, but it does not explicitly

indicate "failure conditions". Ponnekanti discloses the claimed failure conditions during

the execution of query [Ponnekanti paragraph 0018]. It would have been obvious to one

of ordinary skill in the art of data processing to have combined the cited references

because failure conditions as disclosed by Ponnekanti would have served to enable

MacLeod to process some n-ary nested loop joins more efficiently [Ponnekanti,

paragraph 0010].

Claim 10.

The combination MacLeod and Ponnekanti discloses the elements of claim 9 as

above and furthermore it discloses at least one failure condition of the plurality of failure

conditions indicates the initial query operation and an operation status of the initial

query operation which indicates a failure of the composite query operation [operation

node icon representing an operation in execution plan (operation status), See MacLeod,

Col 8 lines 46-49, Fig 5-6, Fig 10].

Claim 11.

The combination MacLeod and Ponnekanti discloses the elements of claim 10 as

above and furthermore it discloses each other failure condition of the plurality of failure

conditions indicates a series of the initial query operation and at least one of the plurality

of subsequent query operations, and an operation status of the at least one of the

plurality of subsequent query operations which indicates a failure of the composite

query operation [operation node icon representing an operation in execution plan

(operation status), See MacLeod, Col 8 lines 46-49, Fig 5-6].

Claim 12.

The combination MacLeod and Ponnekanti discloses the elements of claim 9 as

above and furthermore it discloses:

a) executing the initial query operation [MacLeod Fig 5-6, Fig 10];

b) determining an operation status of the initial query operation [operation node

icon representing an operation in execution plan (operation status), See MacLeod, Col 8

lines 46-49, Fig 5-6];

c) determining, on the basis of the operation status and the plurality of failure

conditions, whether failure of the composite query operation occurred[conditions failed,

Ponnekanti paragraph 0018]; and

d) if no failure of the composite query operation occurred [condition succeeds,

Ponnekanti Fig 8, paragraph 102]:

i) determining the next operation to be executed from the plurality of subsequent

query operations using the selection logic [Ponnekanti Fig 8, paragraph 100];

ii) executing the next query operation [Ponnekanti Fig 8, paragraph 100];

iii) determining an operation status of the next query operation [operation node

icon representing an operation in execution plan (operation status), See MacLeod, Col 8

lines 46-49, Fig 5-6];

iv) determining, on the basis of the operation status and the plurality of failure

conditions, whether failure of the composite query operation occurred [Ponnekanti Fig 8,

paragraph 100,102]; and

v) repeating step d) for at least one other of the plurality of subsequent query

operations [Ponnekanti Fig 8, paragraph 100,102].

Claim 14.

The combination MacLeod and Ponnekanti discloses the elements of claim 12

as above and furthermore it discloses wherein step b) and step iii) comprise determining

whether the respective query operation completed successfully [succeeds, Ponnekanti

Fig 8, paragraph102,105] .

Claim 16.

The combination MacLeod and Ponnekanti discloses the elements of claim 12 as

above and furthermore it discloses, wherein step b) and step iii) comprise determining

whether the respective query operation completed successfully and, if the respective

query operation completed successfully [succeeds, Ponnekanti Fig 8,

paragraph102,105]:

completing execution of the composite query operation [query execution,

Ponnekanti Fig 8, paragraph102,105].

Claim 17.

The combination MacLeod and Ponnekanti discloses the elements of claim 9 as

above and furthermore it discloses, wherein the initial and each subsequent query

operation is an SQL statement [SQL, Ponnekanti paragraph 0012].

Claim 51.

MacLeod discloses:

A data processing system comprising [processing unit, See MacLeod, Col 7 lines

23-25]:

a data repository [data stored, database, See MacLeod, Col 5 lines 64-66]; and

a composite query operations manager residing in memory for managing

execution of query operations in the data processing system, the composite query

operations manager being configured for [management system, execution plan, See

MacLeod, Col 6 lines 10-15]:

receiving a request to perform a composite query operation defining an initial

query operation and a plurality of subsequent query operations to be executed against

the data repository [MacLeod, Col 1 lines 65-67, Col 7 lines 28-35, 43-45, Fig 10];

retrieving selection logic defining a next query operation of the composite query

operation to be executed [next operation, MacLeod Fig 10];

managing execution of the initial query operation and the plurality of subsequent

query operations on the basis of the selection logic and the plurality of failure conditions

[second query (subsequent query) Col 7 lines 63-65, Col 8 lines 1-5,15-25, Fig 5, Fig

10].

MacLeod discloses the elements of claim 51 as above, but it does not explicitly

indicate "failure conditions". Ponnekanti discloses the claimed failure conditions during

the execution of query [Ponnekanti paragraph 0018]. It would have been obvious to one

of ordinary skill in the art of data processing to have combined the cited references

because failure conditions as disclosed by Ponnekanti would have served to enable

MacLeod to process some n-ary nested loop joins more efficiently [Ponnekanti,

paragraph 0010].

Claim 54.

MacLeod discloses:

A data structure residing in memory, comprising [data structure, Col 3 lines 44-

46]:

at least one composite query operation component including [MacLeod Col 6

lines 10-13]:

a plurality of query operations adapted for defining a composite query operation

to be executed against a data repository of a data processing system [MacLeod Col 6

lines 10-15];

selection logic defining a next query operation of the composite query operation

to be executed [next operation, query, MacLeod Fig 10]; and

a plurality of failure conditions for determining when a failure of the composite

query operation occurs; and

at least two implementation schemas for the composite query operation, each

defining a different order of execution of at least two different query operations required

to perform the composite query operation [execution plans (implementation schema),

Col 1 lines 65-67-Col 2 line 1-5, Col 6 lines 10-15, Fig 4].

Subject matters of claims 33-36, 38, 40-41 are rejected in the analysis above in

claims 9-12, 14, 16-17 and these claims are rejected on that basis.

8.   Claims 13 , 15 , 37, 39, are rejected under 35 U.S.C. 103(a) as being

unpatentable over MacLeod in and Ponnekanti further in view of Krishna).

Claim 13.

The Combination of MacLeod and Ponnekanti discloses the elements of claim

12 as above and furthermore it discloses wherein step c) and iv) comprise, if failure

occurred: completing execution of the composite query operation [condition failed,

Ponnekanti Fig 8, paragraph 101]; and

returning a failure code as completion status of the composite query operation

indicating a failure condition from the plurality of failure conditions [condition failed,

Ponnekanti Fig 8, paragraph 101].

The Combination of MacLeod and Ponnekanti disclose the elements of claim 13

as above but it does not explicitly indicate "return code". Krishna discloses the claimed

"return code" (both successful and unsuccessful execution of a query ends with a

return) [see Krishna Fig 3, Col 6 lines 40-45].

It would have been obvious to one of ordinary skill in the art of data processing to

have combined the cited references because "return code" as disclosed by Krishna

would have enabled MacLeod to provide an alternative for un-nesting queries with equi-

join predicate and a "COUNT" aggregate between the nested blocks [Krishna Col 3

lines 19-22].

Furthermore it would enable the outer join operator to permit the application of

different predicates to the join tuples and anti-join tuples [Krishna Col 3 lines 37-39].

Claim 15.

The Combination of MacLeod and Ponnekanti discloses the elements of claim 12

as above and furthermore it discloses wherein step b) and iii) comprise determining, on

the basis of a return code received upon completion of the respective query operation,

whether the respective query operation completed successfully (both successful and

unsuccessful execution of a query ends with a return) [see Krishna Fig 3, Col 6 lines 40-

45].

Subject matters of claims 37 and 39 are rejected in the analysis above in claims

13 and 15, and these claims are rejected on that basis.

Conclusion

Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Emeka Ebirim whose telephone number is 571-272-

3994. The examiner can normally be reached on 8:30pm - 5:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Hosain Alam, can be reached on 571-272-3978. The fax phone number for

the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the

Patent Application Information Retrieval (PAIR) system. Status information for

published applications may be obtained from either Private PAIR or Public PAIR.

Status information for unpublished applications is available through Private PAIR only.

For more information about the PAIR system, see http://pair-direct.uspto.gov. Should

you have questions on access to the Private PAIR system, contact the Electronic

Business Center (EBC) at 866-217-9197 (toll-free).

Emeka Ebirim
Examiner
Art Unit 2166

May 12, 2006

KHANH B. PHAM
PRIMARY EXAMINER